AMENDMENTS TO THE CLAIMS

Please amend claims 1, 8, 22, 25 and 27, such that the current status of the claims is as follows:

1. **(Currently Amended)** A workflow management system for hosting process-based tasks and decisioning, the workflow management system comprising:

a compiled program kernel containing multiple differentiated tasks defined prior to runtime setup as separate functions with the compiled program;

a graphical interface having a list of geometric shapes and a workspace, each geometric shape being an abstracted object-based representation of functions within the compiled program kernel, the workspace for organizing and linking multiple geometric shapes in an ordered arrangement of objects, the ordered arrangement of objects corresponding to an order in which the multiple differentiated tasks are performed by the compiled program kernel;

a database for storing the arrangement of objects as a checklist as well as for storing entry conditions that are associated with each of the multiple differentiated tasks; and

a data dictionary defining discrete data elements and data relationships, wherein the contents of the data dictionary are specific to a selected industry, and wherein the entry conditions are evaluated by the compiled program kernel with respect to each of the multiple differentiated tasks such that a particular one of the multiple differentiated tasks is performed only if all of the entry conditions associated with that particular one of the multiple differentiated tasks evaluate to true.

2. **(Original)** The workflow management system of claim 1, further comprising:

administrative tools for accessing a stored checklist, the administrative tools capable

of altering parameters associated with each geometric shape in the stored checklist.

3. **(Original)** The workflow management system of claim 1, wherein multiple checklists may be stored in the database.

4. **(Original)** The workflow management system of claim 1, wherein the graphical interface permits dynamic alteration of the ordered arrangement of objects in the stored checklist without restarting the system and without recompiling the compiled program kernel.

5. **(Original)** The workflow management system of claim 1, wherein the graphical interface is web-enabled, such that a remote user can access the graphical interface to modify the ordered arrangement of objects in the stored checklist.

6. **(Original)** The workflow management system of claim 1, further comprising:

an automated messaging system for communicating action items with registered users in the system, the messaging system being Internet-based.

7. **(Original)** The workflow engine of claim 6, wherein the automated messaging system includes electronic mail.

8. **(Currently Amended)** A workflow system for programmatically managing dynamic workflow processes, the workflow system comprising:

a workflow engine for performing task list processing, the workflow engine being a software component containing a plurality of discrete functions <u>defined prior to runtime setup</u>;

a workflow designer for configuring task lists, the workflow designer having an

object-based interface for drag-and-drop creation of task lists, the workflow designer having a display window divided into a function list and a workspace, the function list containing multiple symbols, each symbol corresponding to at least one of the plurality of discrete functions within the workflow engine, the workspace providing a graphical area for assembly of ordered task lists, the workflow designer allowing for assembly of ordered tasks by dragging and dropping one of the multiple symbols into the workspace, the workflow designer provides graphical links for assembling an ordered task list from multiple discrete symbols, and wherein the workflow designer allows entry conditions to be defined and associated with any of the plurality of discrete functions, wherein each entry condition is evaluated by the workflow engine with respect to each of the plurality of discrete functions such that a particular one of the plurality of discrete functions is executed by the workflow engine only if all of the entry conditions associated with that particular one of the plurality of discrete functions evaluate to true; and

a data dictionary defining discrete data elements and data relationships that are associated with each of the plurality of discrete functions of the workflow engine, wherein the contents of the data dictionary are specific to a selected industry;

wherein the workflow engine performs discrete functions for which all associated entry conditions evaluate to true in an order determined by the ordered task list to render a financial offer decision to a remote user.

9. **(Original)** The workflow system of claim 8, wherein the workflow designer is Internet-based and wherein the function list and the workspace are accessible using an Internet browser.

10. **(Original)** The workflow system of claim 8, further comprising:

a workflow setup utility for configuring parameters within the checklist.

11. **(Original)** The workflow system of claim 10, wherein the workflow setup utility is web-enabled.

12. **(Original)** The workflow system of claim 8, further comprising:

a messaging system for programmatically prompting a user to take action.

13. **(Original)** The workflow system of claim 12, wherein the messaging system generates a digital message.

14. **(Original)** The workflow system of claim 12, wherein the messaging system forwards a document to the user for review and action.

15-20. **(Canceled)**

21. **(Previously Presented)** The workflow management system of claim 1, wherein the contents of the data dictionary are specific to the lending industry.

22. **(Currently Amended)** A system for programmatically rendering a process-based decision, the system comprising:

administrative tools for creating process categories and checklists associated with each process and for modifying entry conditions and selection criteria associated with discrete tasks defined prior to runtime setup and available in each checklist, wherein the entry conditions define rules that govern whether or not each of the discrete tasks is performed;

a decision database for storing the process categories, the checklists, the entry conditions and the selection criteria;

a workflow engine for automatically processing input from a remote user and generating an instant decision based on the checklist, the entry conditions and the selection criteria associated with the checklist, and the processed input associated with the entry conditions and the selection criteria, wherein the workflow engine is capable of securely transmitting the instant decision to the remote user, and wherein the workflow engine is capable of brokering communications between the remote user and a process administrator associated with the instant decision;

a dynamic data dictionary formatted in XML for defining data elements and data relationships specific to a selected industry, wherein the dynamic data dictionary provides a dynamic fetch and store interface with the decision database, and wherein the dynamic data dictionary is configured to provide, translate and modify data presentation with respect to both the remote user and the workflow engine; and

a messaging system for routing two-way communications between the remote user and the process administrator, the messaging system providing a digital record of programmatic transactions.

23. **(Previously Added)** The system of claim 22, further comprising:

a user interface for entering user information.

24. **(Previously Added)** The system of claim 22, wherein the entry conditions and the selection criteria associated with the checklist are modified and new checklists are created dynamically without restarting the system.

25. **(Currently Amended)** A method for workflow processing and programmatic decision-making based on object-based processes stored in memory, the method comprising:

defining a plurality of differentiated tasks at a pre-runtime software design stage;

receiving input from a remote source;

determining programmatically an input type according to the received input using sets of entry conditions that are associated with each of ~~a~~ the plurality of differentiated tasks and a data dictionary that defines data elements and data relationships used to process the entry conditions, wherein contents of the data dictionary are specific to a selected industry, and wherein each entry condition of the sets of entry conditions is based upon one or more of the data elements and the data relationships defined by the data dictionary;

retrieving automatically a stored process checklist from a decision database according to the input type using the data dictionary as an interface between the stored process checklist and the sets of entry conditions and as an interface between the entry conditions and both the data elements and the data relationships;

processing programmatically the received information utilizing one or more of the plurality of differentiated tasks based on the entry conditions associated with the stored process checklist, wherein each set of entry conditions is evaluated with respect to each of the plurality of differentiated tasks such that a particular one of the plurality of differentiated tasks is performed only if all of the entry conditions associated with that particular one of the plurality of differentiated tasks evaluate to true;

rendering an automatic decision based on the processed received information; and

communicating programmatically the automatic decision to the remote source.

26. **(Previously Added)** The method of claim 25, wherein the step of processing comprises:

querying the data dictionary for a set of data elements and data relationships related to the received input;

receiving a response containing the set of data elements and data relationships related to received input; and

evaluating the received response and the received input according to the entry conditions associated with the stored process checklist.

27. **(Currently Amended)** The method of claim 25, wherein before receiving, the method comprises:

creating a process checklist using an administrative utility, the administrative utility having an object-based, graphical interface driven by the data dictionary wherein an authorized user creates a workflow process by dragging and dropping the defined tasks into a workspace and linking tasks into an interrelated order;

configuring the set of entry conditions using the administrative utility, wherein each set of entry conditions is associated with one of the plurality of differentiated tasks in the process checklist using the administrative utility, and wherein each set of entry conditions define rules that govern whether or not the associated one of the plurality of differentiated tasks is performed as a function of input received; and

storing the process checklist in a decision database.